CORE
Metadata, citation and similar papers at core.ac.uk
Provided by UC Research Repository

Seamless Handoff Between 802.11b and CDMA2000 Networks

November 7, 2003

Kieran Molloy
Department of Computer Science
and Software Engineering
University of Canterbury
Christchurch, New Zealand

Supervisor: Ray Hunt

Abstract

With the deployment of 3G networks by telecommunication companies, and the gradual implementation of wireless networks, seamless handoff between the networks becomes an increasingly desirable facility. Seamless handoff between these systems is possible, although support for seamless handoff is somewhat lacking and research mainly directed toward IPv6 networks.

Contents

1 Introduction

The methods which can be used to connect to the Internet is becoming more diverse. It is possible now to connect with not only the traditional modem, but also through local Ethernet mechanisms, as well as wireless interfaces. Each of these mechanisms has its own strengths and weaknesses. This paper looks at two recent interfaces – 802.11b (WiFi) and CDMA2000 (3G cellular) interfaces for connection to the Internet. 802.11b provides reasonably high speed (up to 11Mb/s) communication, however is somewhat limited in range. CDMA2000 provides slow speed high cost communications, however coverage is nationwide and reliable. These technologies are complimentary. By combining them, it is possible to achieve high speed communications within certain hot-spots, while still maintaining reliable nationwide coverage while outside of the hot-spots.

Seamless handoff describes the ability to switch between different technologies without requiring any input from the user. This allows the best connection available to be used, while not constraining the user to the drawbacks of the connection. It also provides the ability to change connections should a better one become available. Switching access technologies should be fast, to provide minimal impact to any of the users sessions. Switching should be invisible to the user, and occur whenever necessary.

There are a number of different standards for implementing a seamless handoff scheme. We have chosen to use the Mobile IP protocol, for IPv4. Our goal was to implement a testbed for investigating the feasability and properties of seamless handoffs, between wireless networks. This paper describes the testbed we have set up for investigating seamless handoffs.

Section 2 describes the access interfaces being used in more detail, while section 3 details previous research on the topic. Section 4 explains the mobile IP protocol in more detail. Sections 5 and 6 describe the testbed we have set up, as well as the steps involved in setting it up and getting it working. Section 7 details implications of security in the network, while section 8 describes the difficulties faced in implementing the testbed. Section 9 describes some possibilities for future research in the area of seamless handoff, while section 10 provides a conclusion.

2 Access Interfaces

In this paper we look at seamless handoff between 802.11b and 3G CDMA networks. Both of these technologies are wireless interfaces for data communication, however they have vastly different characteristics.

2.1 802.11b

This interface is sometimes referred to as WiFi, and provides network connectivity of up to 11Mbps. It was defined by the IEEE, and belongs to the 802.11 family of wireless interfaces. This operates in the unlicensed 2.4Ghz frequency band, providing cheap wireless communication with reasonable range limitations. This can either be run in ad-hoc mode, allowing other 802.11b nodes to communicate directly with each other, or managed mode, where 802.11b nodes communicate through a base station. When run in managed mode, the base station can provide network access to a local wired network, or even the Internet.

We are using the 802.11b network in managed mode, to provide Internet connectivity to our mobile host. As the 802.11b access point is local, we can determine the configuration of the 802.11b network.

2.2 CDMA2000

CDMA2000 is a 3G cellular data interface, developed by the Third Generation Partnership Project 2 (3GPP2). There are several different forms of CDMA2000. These include CDMA2000 1x, CDMA2000 1xEV-DO, CDMA2000 1xEV-DV, and CDMA2000 3x. These vary in the data bit-rates they provide, ranging from 144 kbps to 4.8 Mbps. We are using CDMA2000 1x, provided by Telecom New Zealand, with a maximum bit-rate of 144 kbps.

The CDMA2000 interface accesses Telecoms cellular network, as such configuration of the CDMA2000 network is managed by Telecom. This prevents any investigation into parameters of the CDMA2000 network interface which affect seamless handoff, such as the ideas presented in [8], however this should ensure the CDMA2000 network configuration is correct and reduce the effort required to connect using this interface.

3 Background

Seamless handoff between 802.11b and UMTS, another 3G standard, was demonstrated by Lucent Technologies[1]. Their implementation focused more on the implementation and interoperability issues which would be faced by telecommunication companies wishing to establish hot spots for high data access in their coverage network. Our research is directed at implementing a small-scale seamless handoff testbed, as would be the case for an enterprise wishing to provide multiple access methodologies for its employees.

The paper [6] investigates the handoff procedure in more detail, focusing on the causes of delay.The paper concludes that the majority of the seamless handoff is spent in detecting location changes. This is not true in our particular case, is movement detection is done by observing the state of the 802.11b interface. If the interface goes down, then our location has changed to the CDMA network, and when the 802.11b interface comes up, our location has returned to the 802.11b home network. This paper also identifies that MTU discovery for the new network, and the TCP exponential backoff algorithm, will introduce additional delays into the handoff.

An alternative to mobile IP is proposed in the paper [7]. While mobile IPv4 requires only modifications to the mobile node, home agent, and foreign agent network stacks, Mobile IPv6 requires modifications to the network stacks of all nodes on the Internet. The system proposed in this paper falls somewhere in between, with modifications required to the home agent, foreign agent, mobile node and all edge routers in the Internet. The proposed system differs from mobile IP in that the home agent informs the edge router of the mobile node's new address, and the edge router can then directly forward packets from the correspondent nodes to the mobile node. This is similar to the mobile IPv6 protocol, however in mobile IPv6, the home agent informs the correspondent node directly of the mobile node's care of address, and the correspondent node then directly forwards the packets to the mobile node.

The paper [8] details the possibility of using Mobile IP for location management in CDMA networks. This would greatly increase the inter-working capabilities of CDMA2000 and 802.11b. If this technique is not implemented, then a mobile roaming protocol must be implemented on the CDMA2000 network to allow for call management and handoff, while mobile IP would be used on top of this roaming protocol to allow for handoff and roaming between the CDMA2000 and 802.11b networks. If CDMA2000 uses mobile IP for location management and handoff, then the 802.11b could use the same infrastructure, reducing the overhead and burden of the handoff management. However, it is up to the telecommunication companies to implement mobile IP for their CDMA2000 handoff management, and is outside the possibilities of this project to investigate.

4 Mobile IP

Several technologies exist to provide seamless handoff between different access technologies[2, 3, 4, 7, 20].

Mobile IP is defined in RFC3344 for IPv4[2], while the Internet draft [3] defines a mobile IP protocol for IPv6. Both mobile IP protocols provide mobility management and call forwarding, by associating the mobile node with a permanent location, called its *home network*. If the mobile node is at its home network, then it can process packets normally, however if the mobile node leaves its home network, then another node will intercept the packets when they arrive at the mobile node's home network and forward them on to wherever the mobile node currently is. When the mobile node is operating at any point other than its home network, it is said to be on a *visited network*

As IPv6 is still under development, it is envisaged the mobile IP protocol can be built into IPv6, allowing for more efficient operation. This will be achieved using a technique called route optimization. In route optimization, the home agent will inform any correspondent nodes of the mobile node's current care-of address, and the correspondent nodes will forward any future packets directly to the mobile node's care-of address. However, most nodes in the Internet are not running an IPv6 stack at present, so implementing an IPv4 seamless handoff protocol enables us to connect to a greater number of hosts in the Internet.

4.1 Mobile IP Addressing

An IP address is used to identify a node's location, and to forward packets to it. However, higher layer protocols, such as TCP and UDP, also use the IP address, along with a port, number to identify data streams. This causes a problem for IP mobility. As the IP address determines how packets are forwarded, the IP address must change when a node moves so that packets continue to be forwarded to it. However, the IP address must remain constant, so any open TCP or UDP sessions are not disrupted. This is the mobile IP paradox.

To combat this, mobile IP associates two IP addresses with the mobile node. The mobile node has an IP address, called its *home address*, which identifies the mobile node when it is connected to its home network. This is a permanent IP address, which is also used in packets to identify the TCP and UDP data streams. When the mobile node moves away from its home network, it must obtain a new IP address so packets can continue to be routed to it. This new IP address is called the *care-of address*, and enables packets to continue to be sent to the mobile node. The combination of two IP addresses, one to identify the TCP and UDP streams, and one to identify the point of attachment, solves the mobile IP paradox.

4.2 Mobile IP Agents

Mobile IP for IPv4 defines up to four different nodes taking part in the communication process. These are

Mobile Node – this is the node for which mobility services are being provided, and is likely to change access points at any given time.

correspondent Node – this is the node we are talking to. This node itself could be mobile, however it must provide a fixed IP address for the communication. If it is a mobile node, this will be its home address, otherwise it will just be the IP address used to contact the node normally.

Home Agent – this is a node which provides call forwarding and location registration functions to the mobile node. The home agent must be located on the same subnetwork as the mobile node's home address. Any packets destined for the mobile node while it is away from home will be intercepted by the home agent, and forwarded on to the mobile node's care-of address.

Foreign Agent – this node provides decapsulation functions to the mobile node, as well as reducing the burden on the IPv4 address space. When the mobile node moves away from its home network, it must obtain a care-of address. This can either be done by using a technique like Dynamic Host Configuration Protocol (DHCP), or the mobile node can listen for the presence of a foreign agent on the new network. If there is a foreign agent on the new network, the mobile node will ask it to provide decapsulation services, and all packets will then be routed through the foreign agent. This means the foreign agent's IP address will be the care-of address. As the foreign agent can serve several mobile nodes at once, this reduces the burden on the already sparse IPv4 address space. However, the Foreign Agent participation is optional. If the foreign agent is not used, then the mobile node is said to be operating in *co-located care-of address* mode.

4.3 Mobile IP Operation

When the mobile node is at home, it can send packets normally using its home address. When the mobile node moves away from its home network, it must obtain a care-of IP address. If a foreign agent is providing mobility services to the mobile node, then the care-of IP address is the IP address of the foreign agent. Otherwise, the mobile node must obtain a care-of address from some other mechanism, such as DHCP. The mobile node then sends a *registration request* to its home agent from its new location, informing the home agent of the mobile node's new care-of address. The home agent then starts intercepting packets which were destined for the mobile node's home address, and tunnels them to the mobile node's new care-of address. This encapsulation changes the packet from having the mobile node's home address as a destination to having the mobile node's care-of address as a destination. When the tunneled packets are received by the mobile node or foreign agent (depending whether the mobile

node is using co-located care-of addressing or not), the packets are decapsulated. This changes the destination address back to the mobile node's home address, ensuring upper layer protocols will continue to function properly.

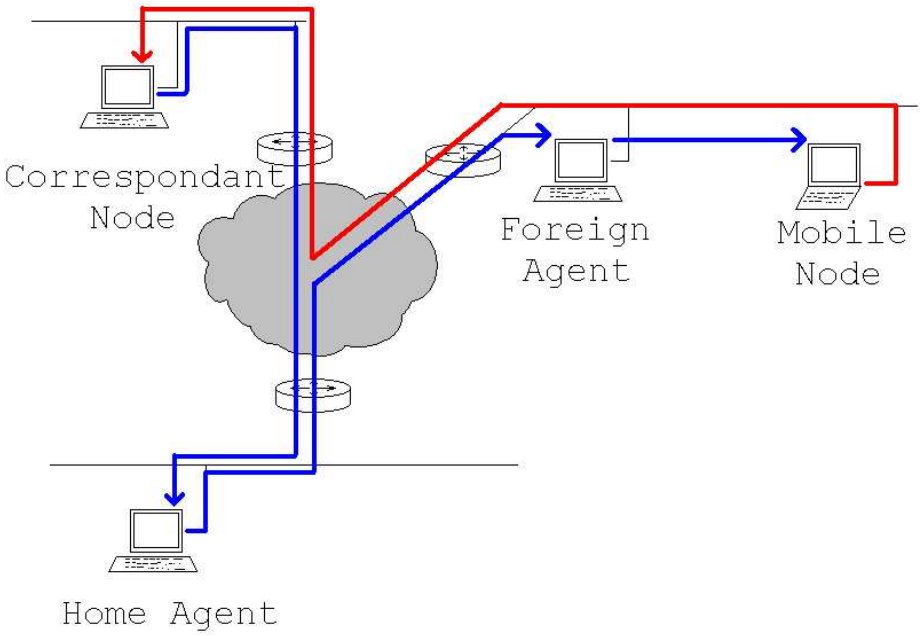

Figure 1: Mobile IP routing using a foreign agent and triangular routing

When the mobile node sends packets back to the correspondent node from a visited network, it can either do one of two things. The packets could be directly routed to the correspondent node, using the mobile node's home address in the source address field. This is called *triangular routing*, and is shown in figure 1. However, this is a technique often used by hackers to hide their location, and routers have begun filtering packets with a source address not matching the source network. To solve this problem, mobile IP also describes a technique called *reverse tunneling*[5], where packets are tunneled back to the home agent, as shown in figure 2. Packets sent will have the care-of address in the source field. This will be able to pass through any intervening routers. When the home agent receives the packets, it decapsulates them and forwards them on to the destination, using the mobile node's home address in the source field.

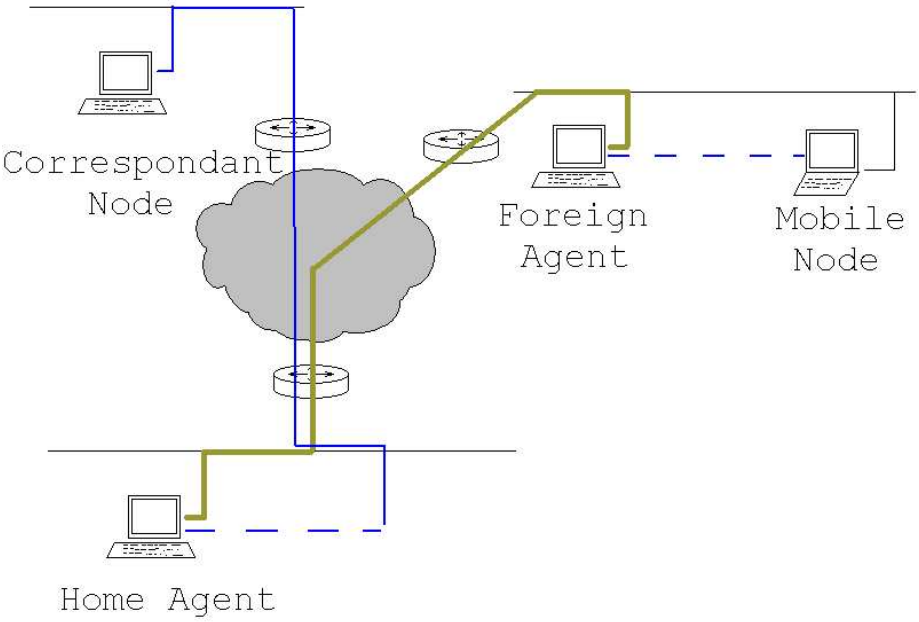

Figure 2: Mobile IP routing using a foreign agent and reverse tunneling

5 The Testbed

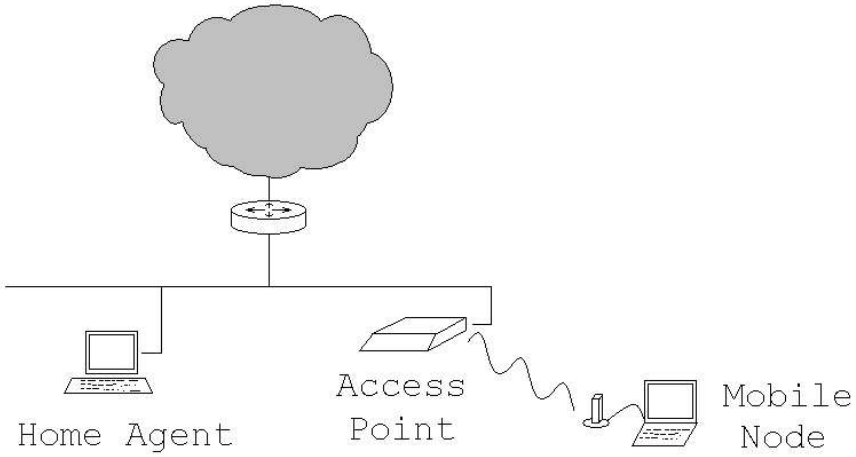

Figure 3: Our testbed, with the mobile node shown at home

Our testbed consists of two computers. A desktop running RedHat 8.0, and Dynamics Home Agent software, provides location registration and call distribution functions. A laptop, also running RedHat 8.0, and Dynamics Mobile Node software, is our test platform. Connection to Telecom New Zealand's CDMA network is provided through a PCMCIA GTran dotSurfer wireless modem, while 802.11b network access is provided through an Orinoco Gold USB card. This can be seen in figure 3. The testbed itself is shown in figure 4

Unfortunately, support for mobile IP in IPv4 networks has been mostly discontinued. Research has focused upon mobile IP for IPv6 networks, as the introduction of mobile IPv6 allows for modifications to the correspondent node's communication stacks, and thus the possibility for extra efficiencies, which is infeasible for mobile IPv4. However, the majority of servers and other nodes for the mobile node to communicate with have not yet implemented the IPv6 communication stack. Thus, any feasible seamless handoff scheme for the current environment should not rely on correspondent nodes supporting IPv6, and we opted for the mobile IPv4 protocol, instead.

There are several implementations of the mobile IPv4 protocol, both commercial[14]

and free[12, 11, 13]. IPUnplugged is a commercial mobile IP implementation, which was used in [1]. This involves the use of specialized home agent hardware.

Of the free mobile IPv4 implementations, we have chosen to use Dynamics, developed by the Helsinki University of Technology[12]. This was done for several reasons, including its ease of implementation, and its support for co-located care-of addressing. Unlike [13], dynamics does not require modifications to the Linux kernel tree. This was desirable as it should reduce the effort required to install the software. Support for Dynamics has been discontinued, with the last update made to it in October 2001. However, this is also the case for [13, 11] as well. Several bugs were noticed while implementing our testbed, especially with regard to co-located care-of address support. However, workarounds were possible for these.

Figure 4: The testbed

6 Setting up the devices

6.1 Home Agent

Setting up the home agent was reasonably straight forward. One of the older machines which was not in use has been utilized for the home agent, as the home agent does not require any intensive CPU operations, and will only be serving one mobile node in our case. Further investigation could be undertaken to determine whether the home agent speed has any adverse affects in single, or multiple, mobile node client scenarios.

The home agent is run on a Pentium II 350Mhz machine, with 256 Mb ram. The home agent connects to the local Ethernet using a 100baseT network adapter. This is connected to the same subnet as the access point through which the mobile node connects when the mobile node is using the 802.11b interface.

To begin setting up the home agent, the hard drive was formatted, and a clean version of RedHat 8.0 was installed. However, the source code included with the RedHat 8.0 kernel did not compile, so the latest 2.4.21 Linux kernel source code was downloaded, and the RedHat kernel updated. The downloaded 2.4.21 kernel could be compiled. Next the correct modules had to be selected. The Dynamics mobile IP software we were using places some restrictions on the possible modules which can be loaded – specifically IP advanced routing and IP-IP tunneling had to be enabled, as well as the various options to get the kernel loaded and running.

Once this was achieved, an IP address needed to be assigned. The home agent was assigned the fixed IP address 192.168.10.56, for which pass-through entries had been added to the gateway firewall. The Dynamics mobile IP implementation was downloaded, unzipped, compiled and installed. This process was completed smoothly. The mobile IP setup was also fairly straightforward. It involved editing the `/etc/dynhad.conf` file, to identify the mobile node. The mobile node's IP address was entered into the file, as well as a shared secret for authentication purposes. Additionally, the home agent had host firewall software enabled by default. This needed to be modified so the home agent would not reject packets sent to it by the mobile node.

Once the home agent software was installed, it was run with the `dynhad --fg --debug` command, so as to display the current state of the system.

6.2 Mobile node

The mobile node is being run on a Compaq laptop, with a Pentium 4 2.4Ghz CPU. This is primarily as the CDMA2000 card is a PCMCIA card, and a compatible hardware system was required.

Setting up the mobile node was more difficult than the home agent. This also involved a clean install of RedHat 8.0, with an updated 2.4.21 kernel. As a USB wireless network connector was being used, the correct USB module had to be enabled for the kernel, as well as the PPP module for the CDMA2000 card, and all the various options for the Mobile IP implementation.

Counter-intuitively, Getting the CDMA2000 card to run had nothing to do with modem dialer programs. Although a username, password and phone number for data services was supplied with the card, entering these into a dialer program did not cause a connection to be established. Instead, getting the CDMA2000 card to work was a matter of using the `pppd` program. A dialer script was created for the `pppd` program, as the username and password still had to be used to set up the connection. This script was called `gtran`.

```
local
default-asyncmap
logfile gtran.log
show-password
user mobile@jamamobile
defaultroute
debug
```

The `gtran` *dialler script*

The script could be run with the command `pppd ttyS3 57600 call gtran`.

Several wireless drivers are built into the Linux kernel, however none which supported the USB card we were using. To get the 802.11b USB card working, independent drivers were downloaded from [9]. There were a few tricks to get this driver working, as it required the firmware for the USB card. The programs `wget` and `curl` were also required, as the USB card drivers uses these programs to fetch the firmware from an Internet site, during the driver build process. These programs were also downloaded, compiled and installed. Compiling the USB drivers additionally required a connection to the Internet. This was achieved through the onboard ethernet interface, which was temporarily enabled for the driver build. Once the drivers were built, they still needed to be run. There was a script bundled with the driver package, however reading through the script, several key lines were commented out. The last three lines of the supplied makefile contained the commands to load the drivers into memory. The `@` needed removing from in front of these lines. Once these lines were uncommented, running the `make load` command would result in the drivers being loaded into memory for the current session. This command must be reissued every time the computer is reset.

Once the drivers are loaded, the wireless interface needs to be set up. This is done using the Linux command `iwconfig`. Among the things to be config-

ured are the mode the card should use (managed or ad-hoc), whether power management should be used, and the WEP security keys. The security keys should be entered corresponding to the way the access point is set up. Here, the security key for index two was used, so the command to set up the encryption parameters is

```
iwconfig wlan0 key [2] s:JJ0ZMTWHTW123
```

The index must be specified for the key to work.

Additionally, after the wireless interface parameters are configured, the interface itself must have certain values assigned. These include the IP address, in this case the IP address was 132.181.8.5. The network mask also needed to be configured, in this case it was 255.255.248.0. The broadcast address was 132.181.15.255, and the default gateway was 132.181.8.254. The local name servers also needed to be included in the `\etc\resolv.conf` file. Once these values were setup, the mobile node should be able to access any local resource through the 802.11b interface, and also should be able to access the Internet through a proxy or any gateway which a regular node on the network can use.

A script was created to run all the setup commands in one batch. This was called `initwlan.sh`

```
ttyS3 57600#!/bin/sh

cd orinoco-usb-0.2.2
make load
cd ..
iwconfig wlan0 key [3] s:jj0zxmwhtw123
iwconfig wlan0 key [3]
ifconfig wlan0 132.181.8.5 netmask 255.255.248.0 broadcast 132.181.15.255
./wlan.sh
cd /etc/ppp/peers
pppd ttyS3 57600 call gtran
```

The `initwlan.sh` *script*

Once the interfaces were set up and configured, the mobile IP implementation also needed to be configured. This was much the same as for the home agent, except it was the `\etc\dynmnd.conf` file which needed editing. The static address of the home agent needed specifying, as well as switching foreign agent requests off. The mobile node software was also configured with the local default gateway. The shared secret needed specifying, as well as whether to request reverse tunneling by default.

After the mobile node software was set up, it could be run with the command `dynmnd --fg --debug`, again to display the state the mobile node software

was in. By issuing a `ifconfig wlan0 down` command, the wireless interface could be brought down. This resulted in the mobile IP software attempting to handoff to the CDMA2000 interface. However, due to a bug in the mobile IP implementation, even though the mobile IP software was running in co-located care-of address mode, the mobile IP software still attempted to connect to a foreign agent. This could be fixed by using the `dynmn_tool` program. Issuing the commands

```
disconnect
tunnel 3
```

forced the mobile IP software to use co-located care-of address mode. This command only needs to be run once, once the mobile IP software is started up. All future changes to interface status result in a seamless handoff occurring.

The mobile node continued to use the local DNS servers, as this allowed it to continue to resolve hosts it was communicating with on the local network. This would not have been possible if it started using foreign DNS servers to resolve local host names, while away from home.

Additionally, two more scripts were created to simulate the wireless interface going down and up, respectively. The `gtran.sh` script simulates the wireless interface going down, while the `wlan.sh` script simulates the wireless interface coming back up.

```
#!/bin/sh

ifconfig ppp0 up
route add -host 132.181.10.56 dev ppp0
ifconfig wlan0 down
```

The `gtran.sh` *script*

```
#!/bin/sh

ifconfig ppp0 down
ifconfig wlan0 up
route add default gw 132.181.8.254
```

The `wlan.sh` *script*

Possibly due to another bug in the mobile IP implementation, bringing the interfaces down and up in the wrong order could result in the registration request being sent using the wrong care-of address. It was discovered that the `ppp` interface, belonging to the CDMA2000 link, had to be brought up before the `wlan` interface was brought down, and also the `ppp` interface had to be brought down before the `wlan` interface was brought up. Failure to do both these operations in this order would result in a seamless handoff in only one direction –

the mobile node could switch access networks once without disrupting communications, but switching in the other direction would result in a large latency as the mobile IP implementation waited for a timer to expire.

Close inspection of this behavior using the packet capturing tool `ethereal` lead to the observation that it was the care-of address which was causing this problem. If the mobile node brought the `wlan` interface up before the `ppp` interface was brought down, then the mobile node would attempt to deregister with the home agent when the `wlan` interface was brought up, only using the `ppp` interface's IP address. The home agent would then reject this deregistration, forcing the mobile node to wait until a retransmission timeout expired.

6.3 Access point

The access point being used was an Orinoco AP2000. The access point has an IP address assigned to it – this is used mainly for configuration purposes. The IP address was obtained using DHCP. Among the other variables to be configured, the network speed was set up to run at 11 Mbps. The network name was "Costa". WEP was enabled, with the key set in index two for both wireless cards of `JJ0ZXMWHTW123`.

There was some difficulty with the access point. After the access point had been working for some time, it suddenly refused to transmit any data over the network, or respond to configuration requests. A reset of the access point did not solve the problem, so the access point had to be re-flashed with a new bios. This solved the problem, however the cause of this bug has yet to be determined.

7 Security

The wireless network was running WEP, to ensure no unauthorized access to the system was permitted. WEP, Wired Equivalent Privacy, is an encryption protocol for 802.11 networks. As 802.11 networks uses radio to transmit data, and radio is inherently a broadcast mechanism, it is possible for external users to observe traffic transmitted on the local network. More importantly, if no security mechanism is implemented, it may be possible for external users to access local resources. For wired networks, access to local resources would normally require a wire link extending from the users computer. However, as 802.11 networks use radio waves, it is possible for external users to access the system from outside the environment where the network operates. WEP prevents this, by encrypting all transmissions over the wireless link, and requiring users to use a shared key to prevent unauthorized access to the local system.

Mobile IP, by default, uses authentication, but not encryption. Mobile IP is a protocol designed to extend the IP protocol to facilitate mobile computing. The IP protocol itself does not provide encryption mechanisms, instead assuming that higher layer protocols will encrypt any sensitive data. As mobile IP is an extension of the IP protocol, it also does not assume any responsibility for ensuring the privacy of data. However, mobile IP must ensure the information it uses to route packets is not falsified. This is done through the use of authentication, and shared secret keys. Whenever a mobile node sends a registration request to the home agent, to update the registered care-of address, the message is authenticated. By default this uses the MD5 hash algorithm.

In our implementation, both the access point through which the mobile node connects, and the home agent itself, were located behind the firewall. This was done for several reasons. Locating the home agent and the access point outside the firewall would expose the home agent and the mobile node to additional security vulnerabilities. Additionally, in order to locate the home agent and access point outside the firewall, the home agent and access point would have to be physically located outside our department, restricting our ability to configure and access these devices.

As the home agent under reverse tunneling decapsulates and forwards packets from the mobile node, locating it behind the firewall could introduce additional vulnerabilities. As both the home agent and the access point are behind the firewall, if an attacker does manage to gain control of either device, then they could use either device to infiltrate the local trusted network. If an attacker was able to convince the home agent that it was the mobile node, then an attacker could use the home agent to inject packets into the local trusted network as if the source was local. Additionally, if an attacker was able to obtain the WEP key for the access point, then they could use the wireless access point as an entry mechanism to the local trusted network.

Although locating the home agent and access point inside the firewall introduces additional security vulnerabilities to the local trusted network, in this case it was felt the benefits outweighed the costs. The home agent only needs to be operational while the system is in use. This reduces the risk the home agent poses. By enabling WEP on the access point, and using a non-default key, this risk is further mitigated.

8 Difficulties

These are some of the difficulties we faced while implementing our system.

Obtaining a decent free distribution of mobile IP was quite difficult. Developing our own implementation of the software was impossible given the time constraints. In the end we settled for the Dynamics implementation, however the Dynamics implementation is a Linux-based format. Given our hardware configuration, a windows based solution would have been preferable. Some difficulty was encountered in setting the required hardware and software up, as several kernel modules needed to be enabled to facilitate the operation of Dynamics, and some external modules were required as well. Setting up the non-standard hardware involved for the wireless communication was incredibly difficult. Once the hardware was installed, it took numerous kernel compiles to ensure the correct modules were enabled while conflicting modules were disabled.

Once the hardware was installed and set up, difficulties were faced in setting up the Dynamics mobile IP implementation. Static IP addresses needed to be obtained for the mobile node and the home agent, as the mobile node must send packets to the home agent as soon as it moves away from home (to do this it must know the home agent's IP address), while the home agent uses the mobile node's home address for security purposes.

As the home agent was located behind our firewall, rules had to be enabled to allow traffic to pass through to the home agent. For a starter, UDP port 434 needed to be opened for incoming traffic for the home agent to allow the mobile nodes requests to reach the home agent. Additionally, if the mobile node was running in reverse tunneling mode, then IP protocol field 4 had to be opened to allow the IP-IP encapsulated packets the mobile node was sending to reach the home agent unhindered. No additional rules were required in the firewall for the mobile node, which behaved as if it were behind the firewall, even when it was not in the home network.

9 Future Work

Many possibilities exist for future work now that the testbed has been established. Some of these include

9.1 Investigation of handoff properties

Now that the testbed is up and running, measurements of handoff properties can begin. Currently, the latency experienced over the cellular network link is much higher than the average latency experienced by the 802.11b network. Preliminary observation puts the 802.11b network round trip time to prominent Internet sites at under 100 ms (MEASURE THIS), while over the CDMA2000 link the latency experienced is not less than 300 ms. When the mobile node moves away from home under the mobile IP protocol, packets must be routed through the home agent. Intuitively, using reverse tunneling, the round trip time for the mobile node should be the sum of the round trip times from the mobile node to the home agent and the home agent to the correspondent node. Using the triangular routing algorithm, the round trip time should be somewhere between the round trip time without mobile IP and the round trip time using the reverse tunneling algorithm. An experiment could be run to determine whether this hypothesis is true.

9.2 Investigation of Reverse Tunneling

Reverse tunneling was designed to increase connectivity for nodes which were blocked by firewalls protecting against IP spoofing. However, this is only necessary if there are firewalls along the IP route which are blocking IP spoofed packets. The reverse tunneling algorithm increases the time taken for packets to be routed from the mobile node to the correspondent node, and also increases the traffic which must flow through the home network. Under multiple mobile nodes, this traffic flowing through the home network could become significant. Therefore, if the mobile node is able to connect to correspondent nodes using the triangular routing algorithm, then it would be beneficial to do so.

An investigation into the increase in connectivity gained by using the reverse tunneling algorithm is therefore a possibility for future research.

9.3 Handoff latency

There is naturally a small delay observed when a handoff occurs. Currently the delay is noticeable, though not enough to disrupt communications, an investigation into this latency could be undertaken. Specifically, any parameters which affect the latency, and the factors which cause and contribute to it. As mentioned in [6], the TCP or UDP MTU discovery, as well as the TCP slow start mechanism, are both likely to contribute to this latency. An investigation into mechanisms to reduce this delay would also be worthwhile.

9.4 Investigation into further handoff schemes

While using reverse tunneling enhances connectivity, it is at the cost of network latency. The additional connectivity may be useful, but in the spirit of the handoff principle, it would be beneficial to only use reverse tunneling where it is appropriate. An investigation could be made into discovering criteria which determine when reverse tunneling is required, so as to implement a handoff scheme which only uses reverse tunneling when required.

Additionally, Mobile IPv6 can provide route optimization capabilities. A seamless handoff scheme which implemented mobile IPv6 would reduce the latency and cost observed by users of the system, although it is at the cost of connectivity to nodes which have not implemented the IPv6 protocol. Thus, a seamless handoff scheme which used mobile IPv6 to connect to other nodes which have implemented the IPv6 protocol stack, while still able to communicate with other nodes using mobile IPv4, would have benefits.

10 Conclusion

Seamless handoff is an enabling technology for the currently available Internet connectivity mechanisms. It enables users to utilize the fastest connection they have available, while being able to switch connection mechanism should a better one become available, or should the current one fail. This would be of great benefit to mobile Internet users, as they could begin an operation from a high speed base station, while continuing the operation as they move around.

We implemented a testbed to investigate seamless handoff. Our testbed enables seamless handoff between 3G CDMA2000 cellular networks, and 802.11b wireless local area networks. Our testbed made use of the freely available Dynamics implementation of mobile IP.

Implementing our testbed proved quite a challenge, with Dynamics being a Linux-based implementation, and the hardware required was non-standard. After much effort, the hardware was brought up and running. The mobile IP implementation was slightly easier to get working, however there are a few bugs in the mobile IP implementation which hampered our efforts. In particular, the co-located care-of address support was slightly lacking, and required several workarounds on our part to get the system to work. However, in the end, seamless handoff between the networks was achieved, allowing the mobile node to begin a transmission on one network, and switch access methodologies back and forth, without disrupting communications.

Based on our experiences, our advice to people who desire a seamless handoff architecture is to at least consider a commercial solution. The cost involved may well prevent the hassles we observed implementing our system.

References

[1] David Benenati, Peretz M. Feder, Nancy Y. Lee, Silvia Martin-Leon, Reuven Shapira: *A Seamless Mobile VPN Data Solution for CDMA2000*, UMTS and WLAN Users*, Bell Labs Technical Journal (October 2002)

[2] C. Perkins, Ed.: *IP Mobility Support for IPv4*, IETF Network Working Group RFC: 3344 (August, 2002)

[3] D. Johnson, C. Perkins, J. Arkko: *Mobility Support in IPv6*, IETF Mobile IP Working Group Internet Draft (30 June, 2003)

[4] A. C. Snoeren and H. Balakrishnan: *An End-to-End Approach to Host Mobility*, Proc. 6th International Conference on Mobile Computing and Networking (Aug. 2000)

[5] G. Montenegro: *Reverse Tunneling for Mobile IP, revised*, IETF Network Working Group RFC: 3024 (January, 2001)

[6] NA Fikouras, K. El Malki, S. R. Cvetkovic, M. Kraner: *Performance analysis of mobile IP handoffs*, IEEE (1999)

[7] Wei Wu, Wen-Shiung Chen, Ho-En Liao, Fongray Frank Young: *A seamless handoff approach of mobile IP protocol for mobile wireless data networks*, IEEE (2002)

[8] Vijay Kumar Garg, Harish Tejwani: *Mobile IP for 3G Wireless Networks*, IEEE (2000)

[9] *http://orinoco-usb.alioth.debian.org*, (August 2003)

[10] Nilanjan Banerjee, Wei Wu, and Sajal K. Das: *Mobility Support in Wireless Internet*, IEEE Wireless Communications (October 2003)

[11] *http://www.monarch.cs.rice.edu/*, (May, 2003)

[12] *http://www.cs.hut.fi/Research/Dynamics/*, (July 2003)

[13] *http://mosquitonet.stanford.edu/mip/*, (May, 2003)

[14] *http://www.ipunplugged.com/*, (April, 2003)

[15] *http://www.3gpp2.org/*, (October, 2003)

[16] Y. Jou: *Developments in Third Generation (3G) CDMA Technology*, QUALCOMM Incorporated, (2000)

[17] Paul Tan: *Recommendations for Achieving Seamless IPv6 Handover in IEEE 802.11 Networks*, (Feb, 2003)

[18] Xavier Pérez Costa, Hannes Hartenstein: *A simulation study on the performance of Mobile IPv6 in a WLAN-based cellular network*, Computer Networks 40 (2002)

[19] Tsuguo Kato, Ryuichi Takechi, Hideaki Ono: *A study on Mobile IPv6 Based Mobility Management Architecture*, Fujitsu Scientific & Technical Journal (June 2001)

[20] Riaz Inayat, Reiji Aibara, Kouji Nishimura, Takahiro Fujita, Yoshihiro Nomura, and Kaori Maeda: *MAT: An End-to-End Mobile Communication Architecture with Seamless IP Handoff Support for the Next Generation Internet*, Springer-Verlag Heidelberg (January 2003)